This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

Fig. 1

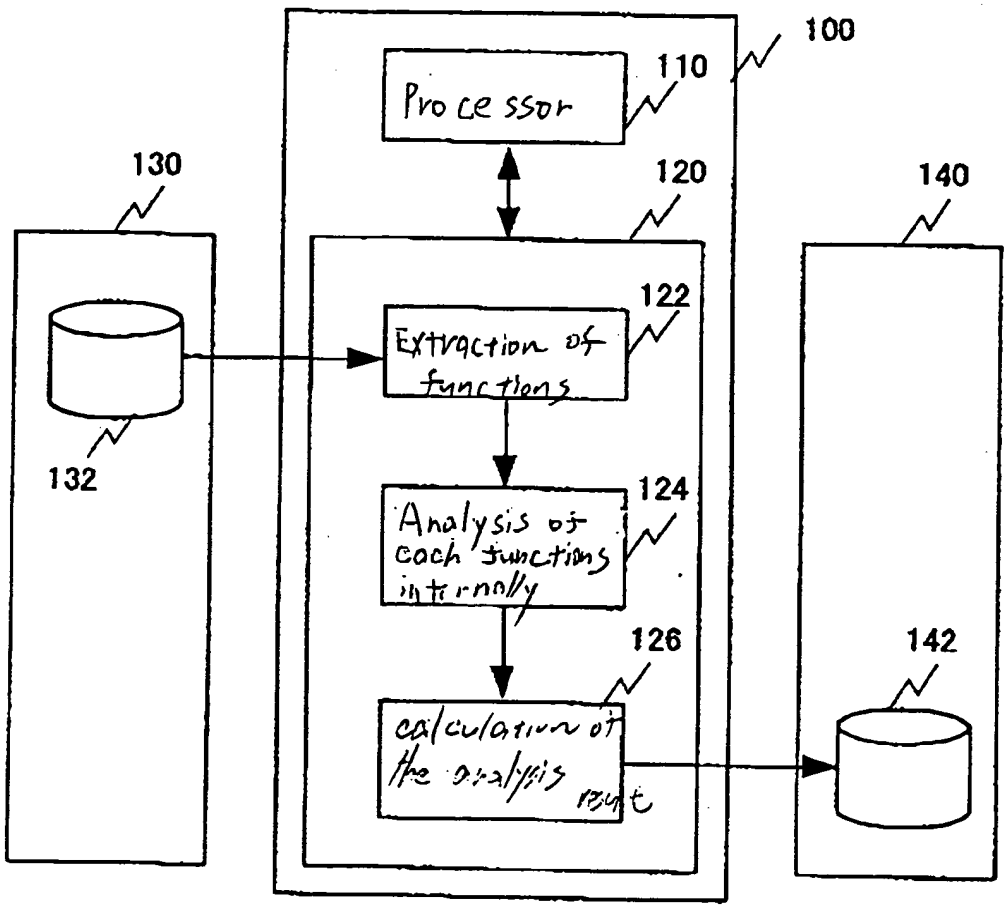
100
110
Processor
120
130
140
122
Extraction of functions
132
124
Analysis of each functions internally
126
142
calculation of the analysis result

Fig.2

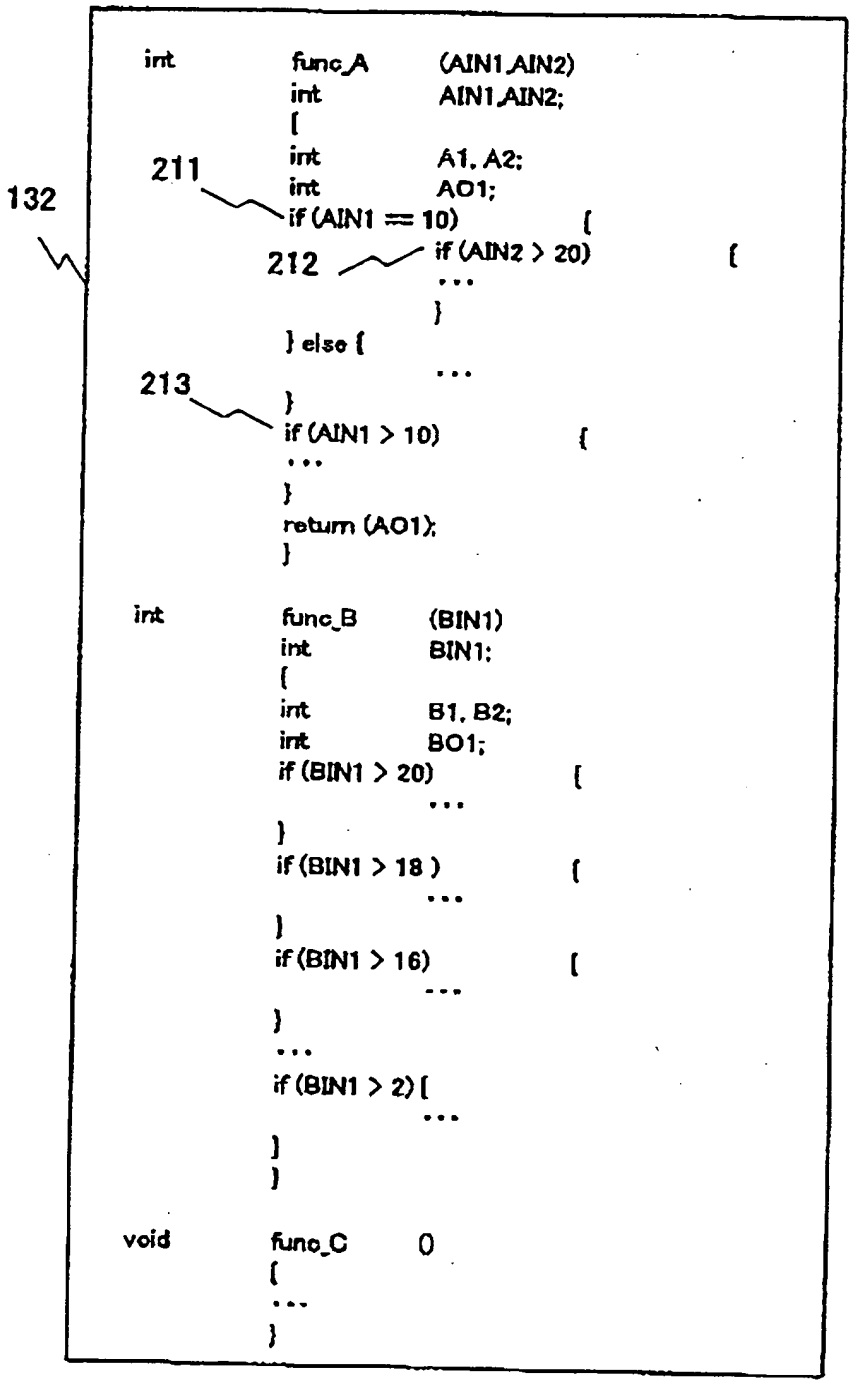
132
211
212
213
int func_A (AIN1,AIN2)
int AIN1,AIN2;
{
int A1, A2;
int AO1;
if (AIN1 == 10) {
if (AIN2 > 20) {
...
}
} else {
...
}
if (AIN1 > 10) {
...
}
return (AO1);
}
int func_B (BIN1)
int BIN1;
{
int B1, B2;
int BO1;
if (BIN1 > 20) {
...
}
if (BIN1 > 18) {
...
}
if (BIN1 > 16) {
...
}
...
if (BIN1 > 2) {
...
}
}
void func_C 0
{
...
}

Fig.3

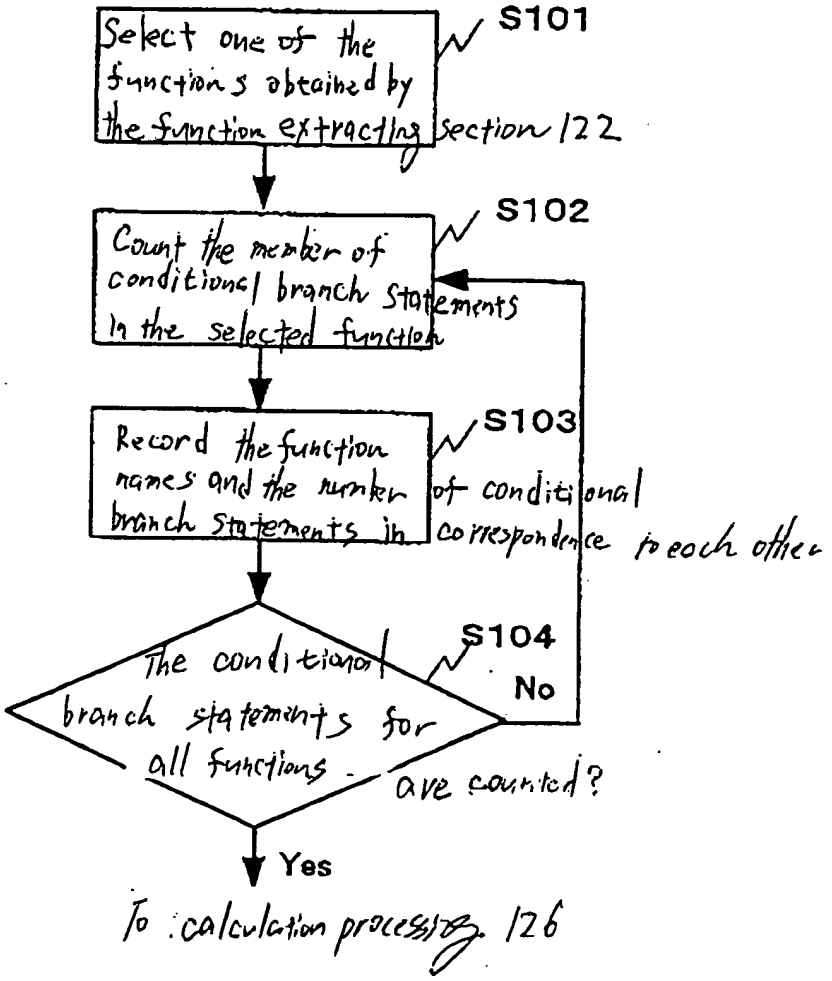

Fig.4

Function name	Number of conditional branch statements
func_A	3
func_B	10
func_C	0

Fig. 5

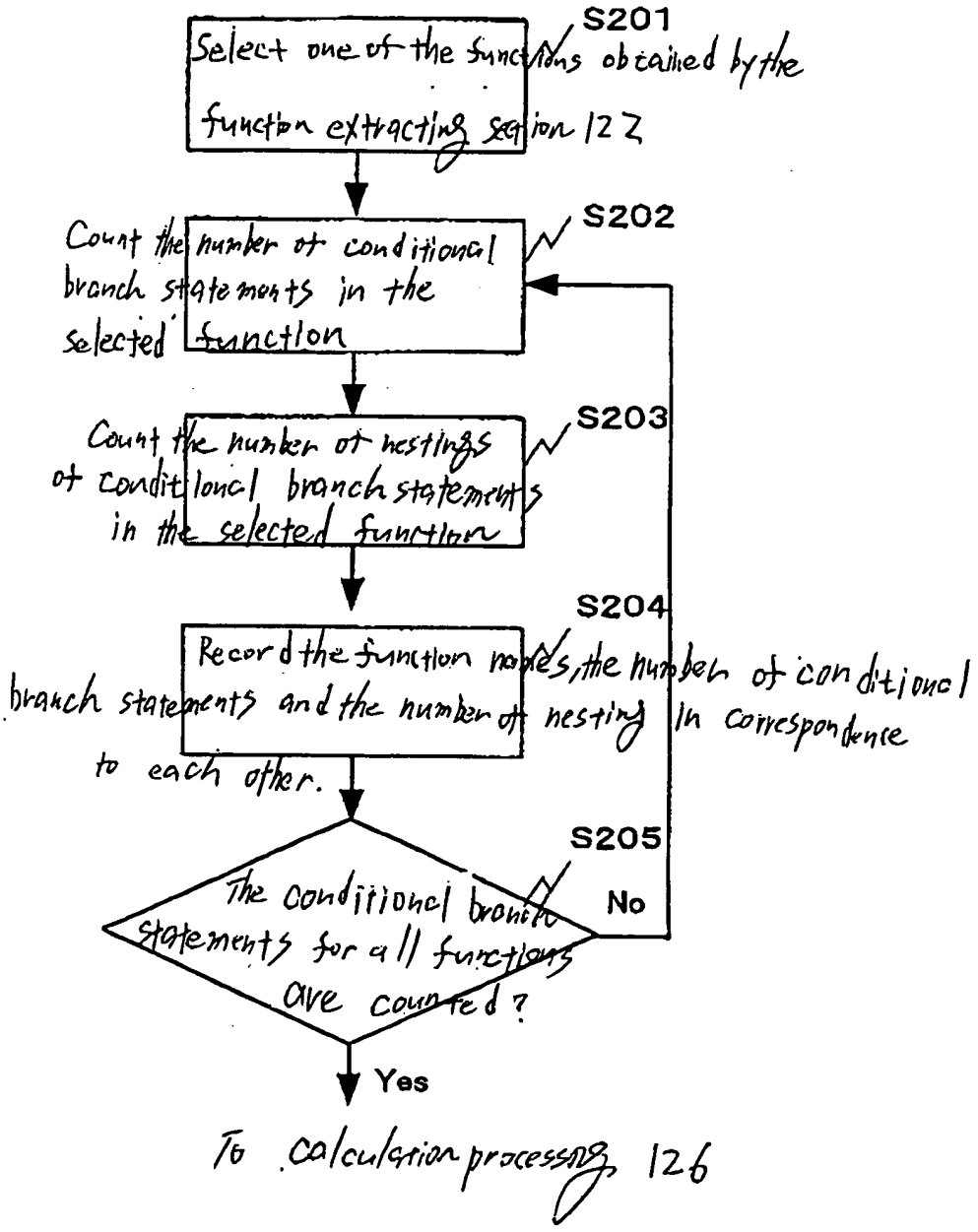
S201
Select one of the functions obtained by the function extracting section 122
S202
Count the number of conditional branch statements in the selected function
S203
Count the number of nestings of conditional branch statements in the selected function
S204
Record the function names, the number of conditional branch statements and the number of nesting in correspondence to each other.
S205
The conditional branch statements for all functions are counted?
No
Yes
To calculation processing 126

Fig. 6

Function name	Conditional branch statement		
	Number	Number of nestings	
		Number of nesting zero	Number of corresponding conditional branch statements
func_A	3	0	2
		1	1
func_B	10	0	10
func_C	0	0	0

Fig. 7

132

```
int         func_A      (AIN1,AIN2)
            int         AIN1,AIN2;
            {
            int         A1, A2;
211         int         AO1;
            if (AIN1 == 10)         {
212             if (AIN2 > 20)          {
                ...
                }
            } else {
                ...
213         }
            if (AIN1 > 10)          {
            ...
            }
            return (AO1);
            }

void        func_D      ()
            {
            int         D1, D2, D3;
            D2 = func_A1 ( D1 );
711         AIN1 = func_A2 ( D2 );
712         ...
            D3 = func_A ( AIN1, D2 );
            }
```

Fig.8

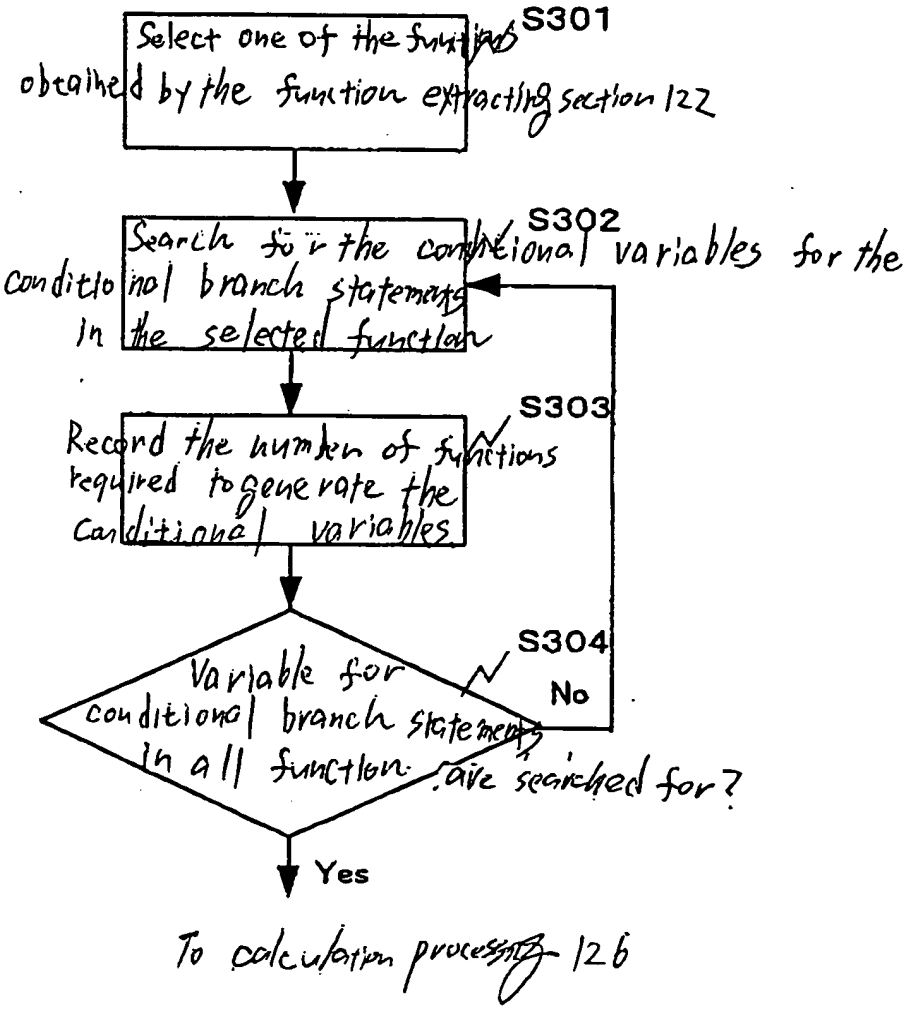

Fig.9

Function name	Conditional branch Statement		
	Number	Variable generation	
		Number of functions required for generation	Number of variables
func_A	3	2	1
		0	1
func_D	0	0	0

Fig. 10

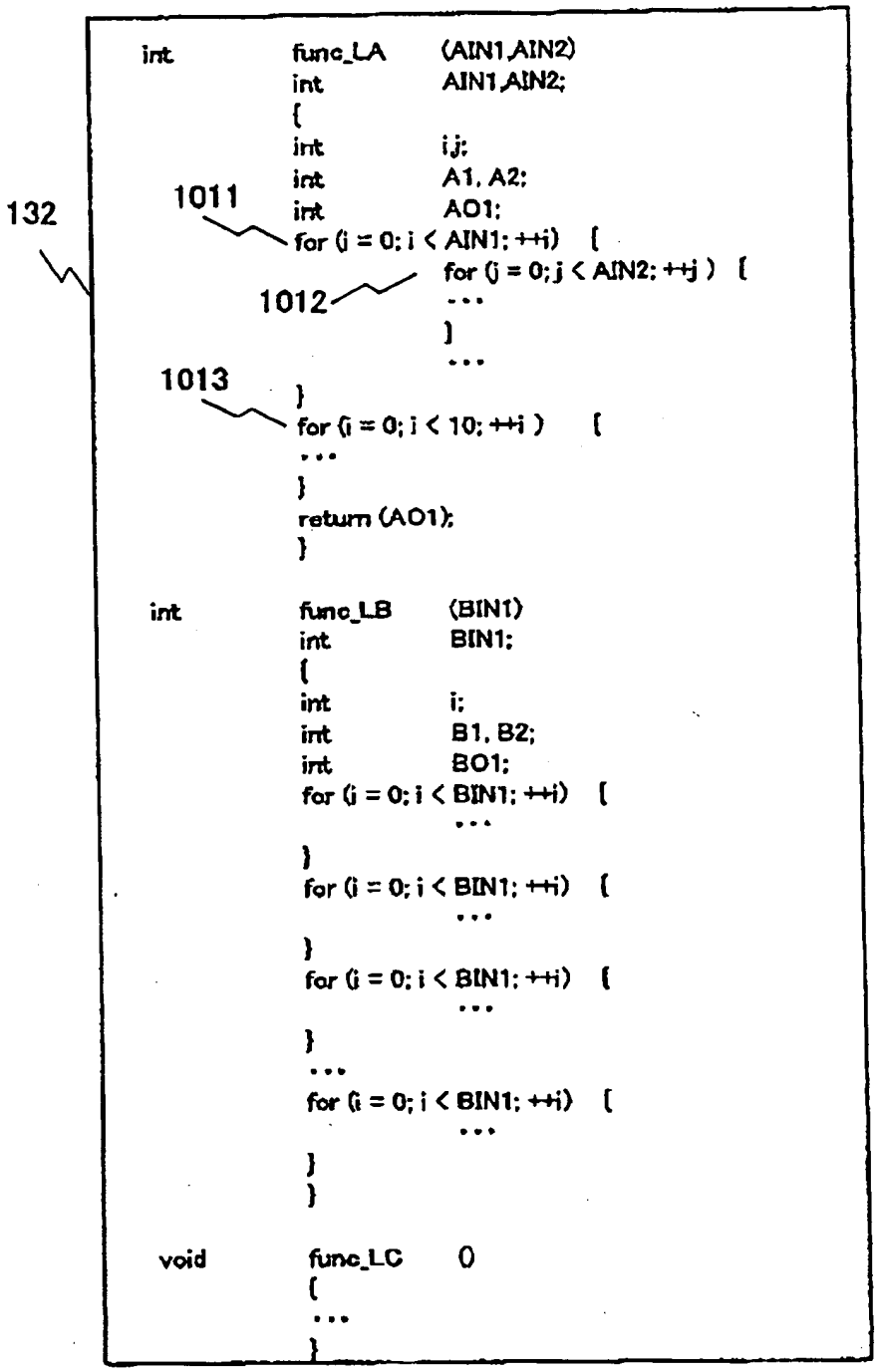
132
1011
1012
1013
int func_LA (AIN1,AIN2)
int AIN1,AIN2;
{
int i,j;
int A1, A2;
int AO1;
for (j = 0; i < AIN1; ++i) {
for (j = 0; j < AIN2; ++j) {
...
}
...
}
for (i = 0; i < 10; ++i) {
...
}
return (AO1);
}
int func_LB (BIN1)
int BIN1;
{
int i;
int B1, B2;
int BO1;
for (j = 0; i < BIN1; ++i) {
...
}
for (j = 0; i < BIN1; ++i) {
...
}
for (j = 0; i < BIN1; ++i) {
...
}
...
for (j = 0; i < BIN1; ++i) {
...
}
}
void func_LC 0
{
...
}

Fig. 11

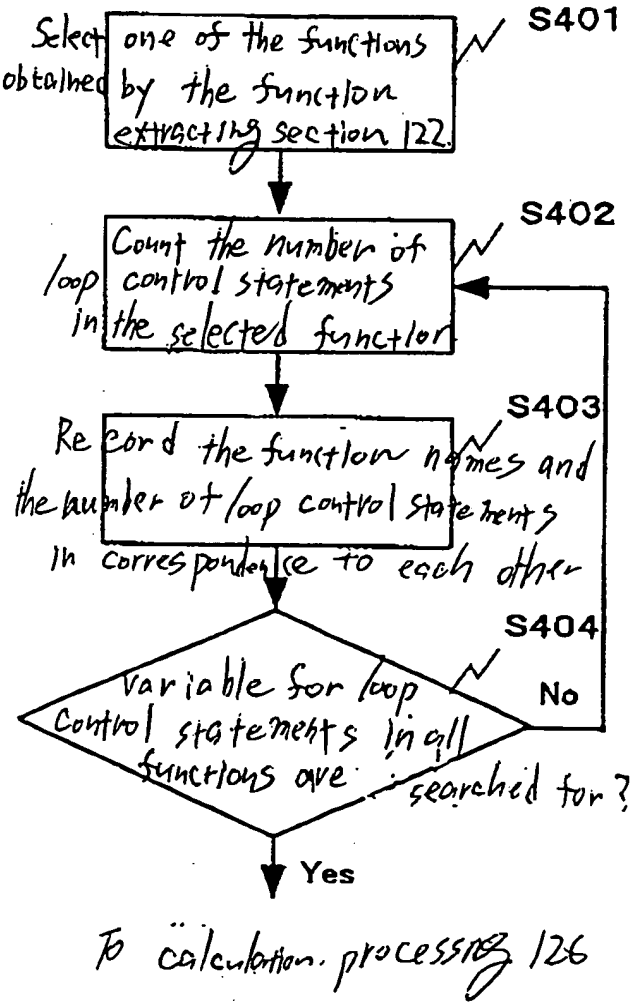

Fig. 12

Function name	Number of loop control statements
func_LA	3
func_LB	10
func_LC	0

Fig. 13

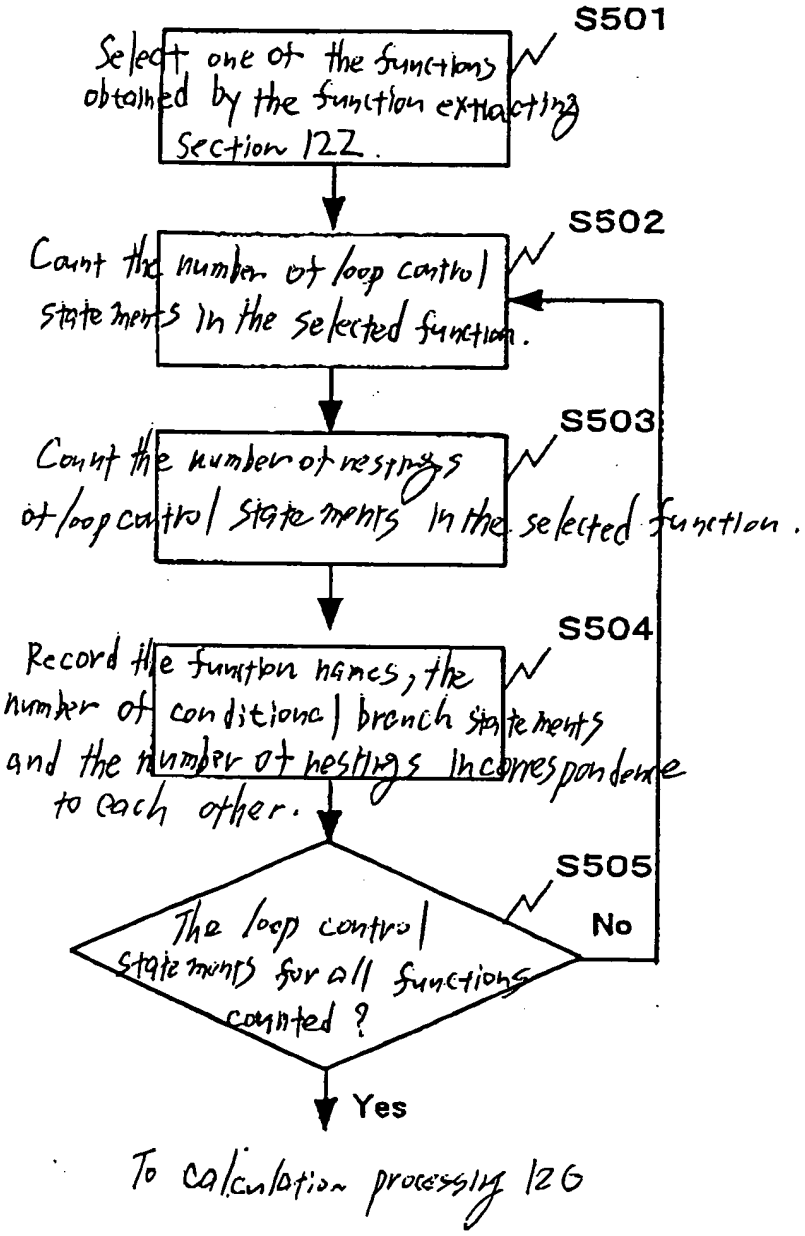
S501
Select one of the functions obtained by the function extracting section 122.
S502
Count the number of loop control statements in the selected function.
S503
Count the number of nestings of loop control statements in the selected function.
S504
Record the function names, the number of conditional branch statements and the number of nestings in correspondence to each other.
S505
The loop control statements for all functions counted ?
No
Yes
To calculation processing 126

Fig. 14

Function name	Loop control statement		
	Number	Number of nestings	
		Number of nesting steps	Number of corresponding loop control statements
func_LA	3	0	2
		1	1
func_LB	10	0	10
func_LC	0	0	0

Fig. 15

```
int         func_LA     (AIN1,AIN2)
            int         AIN1,AIN2;
            {
1011        int         A1, A2;
            int         AO1;
132         for (i = 0; i < AIN1; ++i)  {
1012            for (j = 0; j < AIN2; ++j )  {
                ...
                }
                ...
            }
1013        for (i = 0; i < 10; ++i )    {
            ...
            }
            return (AO1);
            }

void        func_LD     ()
            {
            int         D1, D2, D3;
1511        D2 = func_LA1 ( D1 );
            AIN1 = func_LA2 ( D2 );
1512        ...
            D3 = funcL_A ( AIN1, D2 );
            }
```

Fig. 16

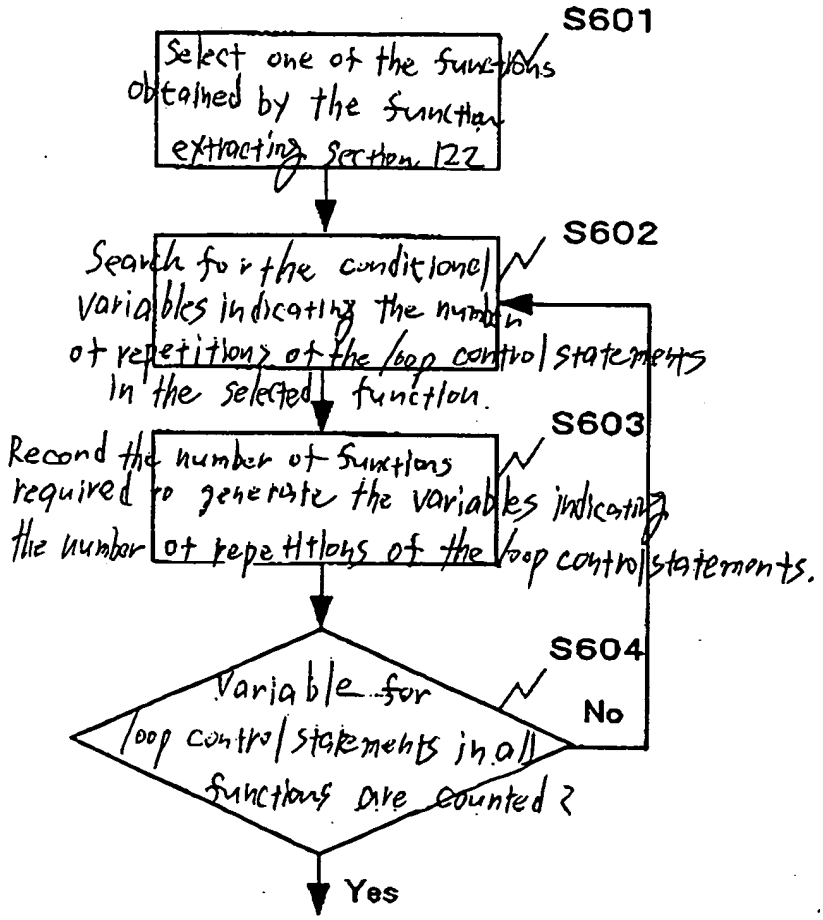

Fig. 17

Function name	Loop control Statement		
	Number	Variable generation for the number of repetitions	
		Number of functions required for generation	Number of variables
func_LA	3	2	1
		0	1
func_LD	0	0	0

Fig. 18

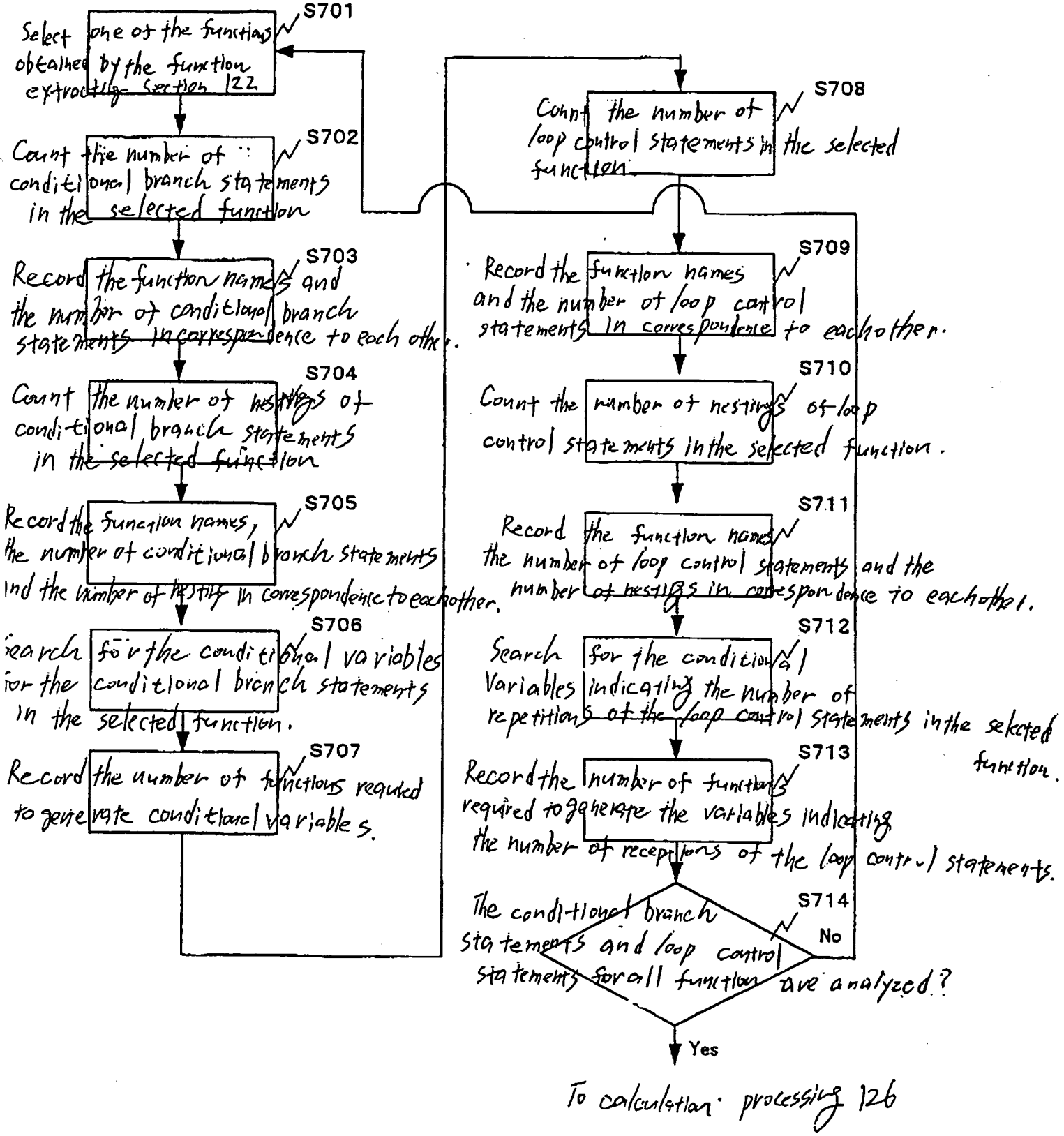
S701
Select one of the functions obtained by the function extracting section 122.
S702
Count the number of conditional branch statements in the selected function
S703
Record the function names and the number of conditional branch statements in correspondence to each other.
S704
Count the number of nestings of conditional branch statements in the selected function
S705
Record the function names, the number of conditional branch statements and the number of nestings in correspondence to each other.
S706
Search for the conditional variables for the conditional branch statements in the selected function.
S707
Record the number of functions required to generate conditional variables.
S708
Count the number of loop control statements in the selected function.
S709
Record the function names and the number of loop control statements in correspondence to each other.
S710
Count the number of nestings of loop control statements in the selected function.
S711
Record the function names, the number of loop control statements and the number of nestings in correspondence to each other.
S712
Search for the conditional variables indicating the number of repetitions of the loop control statements in the selected function.
S713
Record the number of functions required to generate the variables indicating the number of repetitions of the loop control statements.
S714
The conditional branch statements and loop control statements for all functions have been analyzed?
No
Yes
To calculation processing 126

Fig. 14

conditional branch statements

Function name	Conditional branch statement					Loop control statement				
	Number	Number of nestings		Variable generation		Number	Number of nestings		Generation of variables for the number of repetitions	
		Number of nesting stages			Number of variables		Number of nesting stages	Number of corresponding loop control statements	Number of functions required for generation	Number of variables
func_A	3	0	2	2	1	0	0	0	0	0
		1	1	0	1					
func_B	10	0	10	0	1	0	0	0	0	0
func_C	0	0	0	0	0	0	0	0	0	0
func_D	0	0	0	0	0	0	0	0	0	0
func_LA	0	0	0	0	0	3	0	2	2	1
							1	1	0	1
func_LB	0	0	0	0	0	10	0	10	0	1
func_LC	0	0	0	0	0	0	0	0	0	0
func_LD	0	0	0	0	0	0	0	0	0	0

Number of functions required for generation

Fig.20

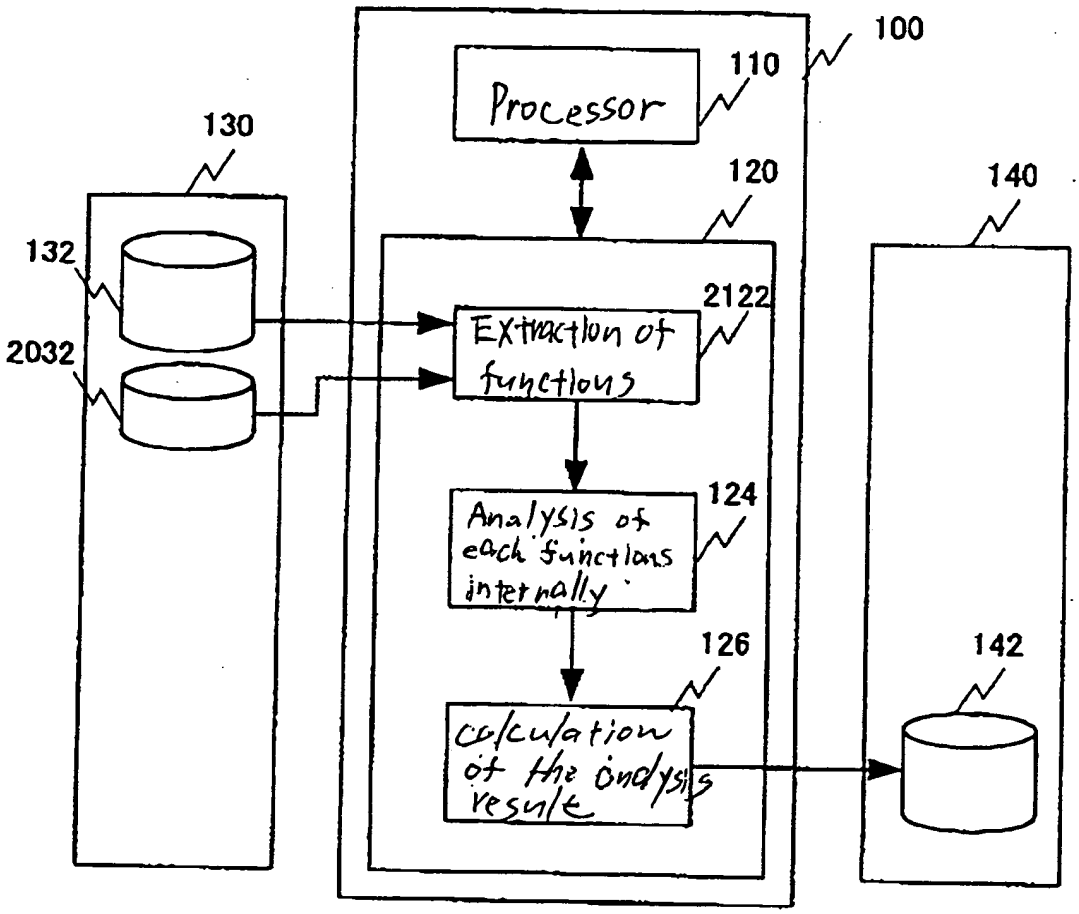

Fig.21

Function name	Conditional branch statement		
	Number	Variable generation	
		Number of functions required for generation	Number of variables
func_A + func_D	3	2	1
		0	1

Fig.22

Function name	Loop control statement		
	Number	Generation of variables for the number of repetitions	
		Number of functions required for generation	Number of variables
func_LA + func_LD	3	2	1
		0	1

Fig.23

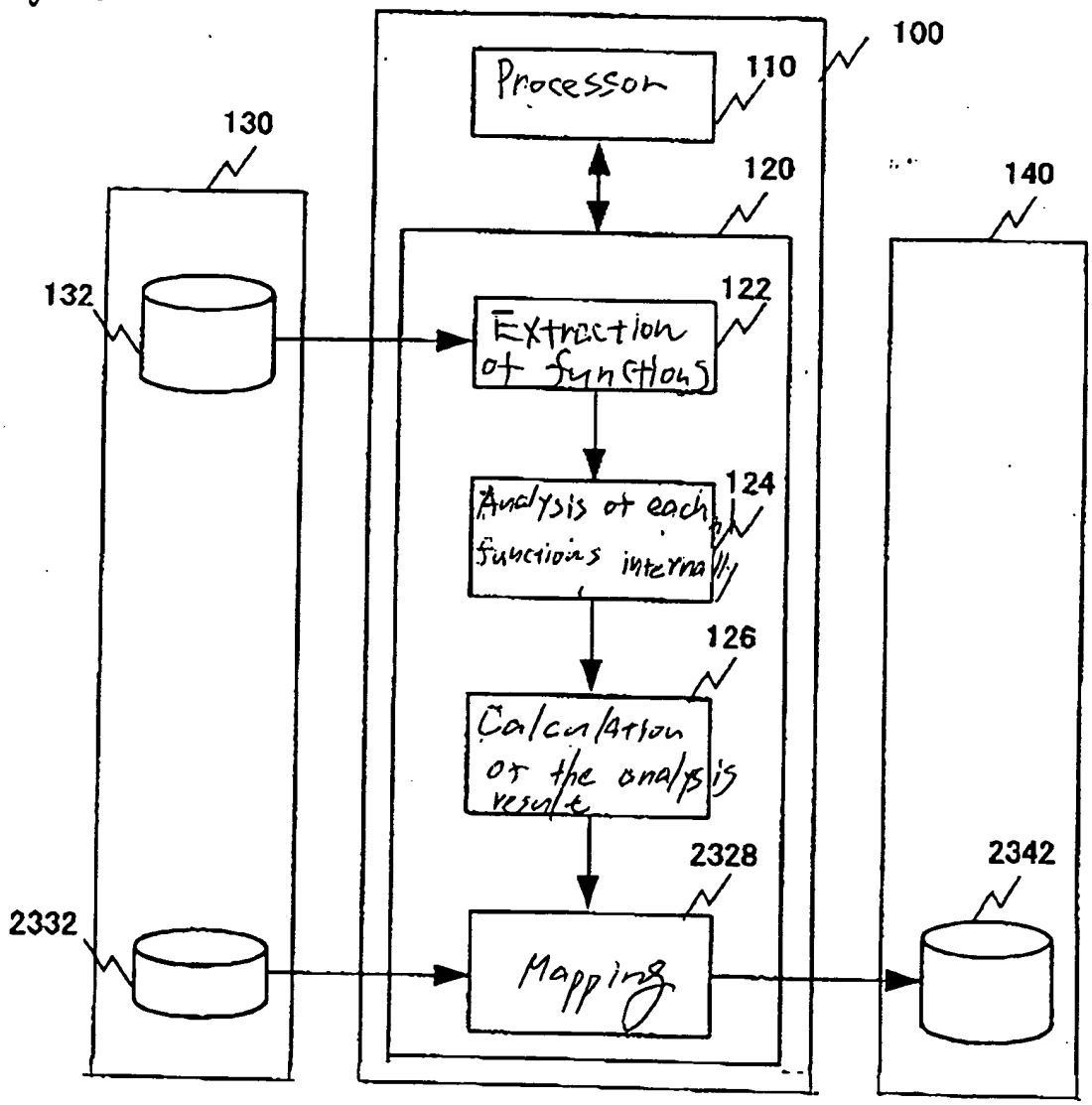

Fig.24

Processing unit	Preferable Number of conditional branch statements
CPU	10 or more
DSP	3 to 9
Dedicated logic	2 or less

Fig.25

Function name	Appropriate processing unit
func_A	DSP
func_B	CPU
func_C	Dedicated logic

Fig.26

Processing unit	Preferable number of loop control statements
CPU	3 to 9
DSP	10 or more
Dedicated logic	2 or less

Fig.27

Function name	Appropriate processing unit
func_LA	CPU
func_LB	DSP
func_LC	Dedicated logic

Fig. 28

Function name	Appropriate conditional branch statement					Appropriate loop control statement				
	number	Number of nestings		variable generation		number	Number of nestings		Generation of variables for number of repetition	
		Number of nesting steps	Number of corresponding conditional branch statements	Number of corresponding conditional branch statements	Number of variables		Number of nesting steps	Number of corresponding loop control statements	Number of functions required for generation	Number of variables
CPU	10 or more	0	1 or more	0	1 or more	3 to 9	0	3 to 9	0	3 to 9
		1	1 or more	1	1 or more		1	0	1	0
		2 or more	1 or more	2 or more	1 or more		2 or more	0	2 or more	0
DSP	3 to 9	0	3 to 9	0	3 to 9	10 or more	0	1 or more	0	2 or less
		1	3 or less	1	3 or less		1	1 or more	1	2 or less
		2 or more	1 or less	2 or more	1 or less		2 or more	1 or more	2 or more	2 or less
Re discrete logic	2 or less	0	3 or less	0	3 or less	3 or less	0	3 or less	0	3 or less

Fig.29

Function name	Appropriate conditional branch statement					Appropriate loop control statement				
	Number	Number of nesting stages	Number of corresponding conditional branch statements	Variable generation: Number of functions required for generation	Variable generation: Number of variables	Number	Number of nesting stages	Number of corresponding loop control statements	Generation of variables for the number of repetitions: Number of functions required for generation	Generation of variables for the number of repetitions: Number of variables
CPU	10 or more (K=8)	0 (K=1)	1 or more	0 (K=1)	1 or more	3 to 9 (K=6)	0 (K=1)	3 to 9	0 (K=1)	3 to 9
		1 (K=2)	1 or more	1 (K=2)	1 or more		1 (K=1)	0	1 (K=1)	0
		2 or more (K=3)	1 or more	2 or more (K=3)	1 or more		2 or more (K=1)	0	2 or more (K=1)	0
DSP	3 to 9 (K=3)	0 (K=1)	3 to 9	0 (K=1)	3 to 9	10 or more (K=6)	0 (K=1)	1 or more	0 (K=2)	2 or less
		1 (K=2)	2 or less	1 (K=2)	3 or less		1 (K=2)	1 or more	1 (K=2)	2 or less
		2 or more (K=3)	1 or less	2 or more (K=3)	1 or less		2 or more (K=3)	1 or more	2 or more (K=3)	2 or less
Dedicated logic	K=6	0 (K=1)	3 or less	0 (K=1)	3 or less	3 or less (K=6)	0 (K=2)	3 or less	0 (K=2)	3 or less

CPU=3+4+2+2=11

DSP=6+4+4+6=20

Dedicated logic. =1+1+6+2+2=12

Fig.30

Function name	appropriate processing unit
func_A	DSP
func_B	CPU
func_C	Dedicated logic
func_LA	CPU
func_LB	DSP
func_LC	Dedicated logic